

APPENDIX

CLASS DIVISION

```
1. filenames = os.listdir('/media/kevin/Kevin/Skripsi/Dataset')
2.
3. categories = []
4. for filename in filenames:
5.     hasil = ''
6.     category = filename.split('.')[0]
7.     if category == 'ct':
8.         categories.append(0) #The categories are set as labels
9.     elif category == 'hm':
10.        categories.append(1)
11.        elif category == 'pk':
12.            categories.append(2)
13.
14. df = pd.DataFrame({
15.     'filename': filenames,
16.     'category': categories,
17.     'hasil' : hasil,
18. })
```

Layer CNN

```
1. from keras.models import Sequential
2. from keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense,
   Activation, BatchNormalization
3. model = Sequential()
4. model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(200,200,
   3)))
5. model.add(MaxPooling2D(pool_size=(2, 2)))
6. model.add(Dropout(0.25))
7. model.add(Conv2D(64, (3, 3), activation='relu'))
8. model.add(MaxPooling2D(pool_size=(2, 2)))
9. model.add(Dropout(0.25))
10. model.add(Flatten())
11. model.add(Dense(96, activation='relu'))
12. model.add(Dropout(0.5))
13. model.add(Dense(3, activation='softmax'))
14. model.compile(loss='categorical_crossentropy', optimizer='Adam',
   metrics=['accuracy'])
```

TRAIN DATASET

```
1 epochs=100
2 batch_size=50
3
4 history = model.fit_generator(
5     train_generator,
6     epochs=epochs,
7     validation_data=validation_generator,
8     validation_steps=total_validate/batch_size,
9     steps_per_epoch=total_train/batch_size
10 )
```

PREDICT TEST DATA

```
1 predict=model.predict_generator(test_generator,steps=np.ceil(total_test/batch_size))
2 test_df['category'] = np.argmax(predict, axis=-1)
```

DISPLAY TEST RESULTS

```
1 sample_test = test_df.head(50)
2 sample_test.head()
3 plt.figure(figsize=(12, 24))
4
5 tpct = 0
6 tnct = 0
7 fpct = 0
8 fnct = 0
9 tphm = 0
10 tnhm = 0
11 fphm = 0
12 fnhm = 0
13 tppk = 0
14 tnpk = 0
15 fppk = 0
16 fnpk = 0
17
18 counter = 0
19 for index, row in sample_test.iterrows():
20     counter+=1
21     filename = row['filename']
22     category = row['category']
23     a = filename.find('.')
24     b = filename[0:a]
25     if b == 'ct':
26         if category == 'crowntail':
27             hasil = 'true';
28             tpct+=1;
29             tnhm+=1;
30             tnpk+=1;
31         else :
32             hasil = 'false';
```



```

33         fpct+=1;
34         if category == 'halfmoon' :
35             fnhm+=1;
36             tnpk+=1;
37         else :
38             fnpk+=1;
39             tnhm+=1;
40
41     elif b == 'hm':
42         if category == 'halfmoon':
43             hasil = 'true';
44             tphm+=1;
45             tnct+=1;
46             tnpk+=1;
47         else :
48             hasil = 'false';
49             fphm+=1;
50             if category == 'crowntail':
51                 fnct+=1;
52                 tnpk+=1;
53             else :
54                 fnpk+=1;
55                 tnct+=1;
56     elif b == 'pk':
57         if category == 'plakat':
58             hasil = 'true';
59             tppk+=1;
60             tnct+=1;
61             tnhm+=1;
62         else :
63             hasil = 'false';
64             fppk+=1;
65             if category == 'crowntail':
66                 fnct+=1;
67                 tnhm+=1;
68             else :
69                 fnhm+=1;
70                 tnct+=1;
71
72
73     img =
74         load_img("/media/kevin/Kevin/Skripsi/haha/data/test/"+filename,
75                 target_size=(200,200)) #rescale(1./255)
76         plt.subplot(10, 5, index+1)
77         plt.imshow(img)
78         plt.xlabel(filename + "\n Predicted is " + "{}".format(category))
79     plt.tight_layout()
80     plt.show()

```

DISPLAY TP, TN, FP, FN

```
1 print('TRUE POSITIF CT : ',(tpct))
2 print('TRUE NEGATIF CT : ',(tnct))
3 print('FALSE POSITIF CT : ',(fpct))
4 print('FALSE NEGATIF CT : ',(fnct))
5 print('\nTRUE POSITIF HM : ',(tphm))
6 print('TRUE NEGATIF HM : ',(tnhm))
7 print('FALSE POSITIF HM : ',(fphm))
8 print('FALSE NEGATIF HM : ',(fnhm))
9 print('\nTRUE POSITIF PK : ',(tppk))
10.print('TRUE NEGATIF PK : ',(tnpk))
11.print('FALSE POSITIF PK : ',(fppk))
12.print('FALSE NEGATIF PK : ',(fnpk))
```

PRECISION RECALL FORMULA

```
1 precisionct = tpct / (tpct+fpct)
2 recallct = tpct/(tpct+fnct)
3 print('CROWNTAIL')
4 print('precision ',precisionct)
5 print('recall ',recallct)
6
7 precisionhm = tphm / (tphm+fphm)
8 recallhm = tphm/(tphm+fnhm)
9 print('\nHALFMOON')
10 print('precision ',precisionhm)
11 print('recall ',recallhm)
12
13 precisionpk = tppk / (tppk+fppk)
14 recallpk = tppk/(tppk+fnpk)
15 print('\nPLAKAT')
16 print('precision ',precisionpk)
17 print('recall ',recallpk)
18
19 print('\nACCURACY ',(tpct+tphm+tppk)/counter)
20 print('\nPRECISSION ',((precisionct+precisionhm+precisionct)/3))
21 print('\nRECALL ',((recallct+recallhm+recallpk)/3))
```



4.16% PLAGIARISM
APPROXIMATELY

Report #13361921

. INTRODUCTION Background The population of ornamental fish is increasing during the Covid-19 pandemic, especially betta fish. In addition to the affordable price, its maintenance is also relatively easy. Betta fish are collected from various groups ranging from children to adults. There are many types of betta fish, but the most common or most famous in Indonesia are crowntail, halfmoon, and plaque. But not everyone can understand the type of betta fish. Therefore, detection of fish species is needed to help people who want to keep betta fish but don't know the type of betta fish. One method that can be used to detect betta fish is the Convolutional Neural Network (CNN). Convolutional Neural Network (CNN) is a type of neural network commonly used in image data. CNN can be used to detect and recognize objects in an image. Broadly speaking Convolutional Neural Network (CNN) not much different from the usual neural network. **4** CNN consists of neurons that have weight, bias and activation functions. The convolutional layer